Quantified Boolean Formulas

QBF (Quantified Boolean Formulas)

Vocabulary	Propositional variables:	$\Sigma = \{p, q, r, \ldots\}$
	Boolean connectives:	$\vee, \wedge, \neg, \rightarrow, \leftrightarrow$
	<u>Quantifiers</u>:	$\exists, \forall$

QBF (Quantified Boolean Formulas)

Vocabulary	Propositional variables:	$\Sigma = \{p, q, r, \ldots\}$
	Boolean connectives:	$\vee, \wedge, \neg, \rightarrow, \leftrightarrow$
	Quantifiers:	$\exists, \forall$

Syntax $\quad \phi : \; p \mid \ldots \mid \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \phi \rightarrow \phi \mid \phi \leftrightarrow \phi$

$\quad\quad \exists p\, \phi \mid \forall p\, \phi \mid \ldots$

QBF (Quantified Boolean Formulas)

Vocabulary	Propositional variables:	$\Sigma = \{p, q, r, \ldots\}$
	Boolean connectives:	$\vee, \wedge, \neg, \rightarrow, \leftrightarrow$
	Quantifiers:	$\exists, \forall$

Syntax

$\phi : \ p \mid \ldots \mid \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \phi \rightarrow \phi \mid \phi \leftrightarrow \phi$

$\quad \exists p\, \phi \mid \forall p\, \phi \mid \ldots$

Semantics

Requires a model $M : \text{FreeVar}(\phi) \rightarrow \{\text{true}, \text{false}\}$

Describes when ϕ holds on M $(M \vDash \phi)$

QBF (Quantified Boolean Formulas)

Vocabulary
Propositional variables: $\Sigma = \{p, q, r, \ldots\}$
Boolean connectives: $\vee, \wedge, \neg, \rightarrow, \leftrightarrow$
Quantifiers: $\exists, \forall$

Syntax
$\phi : \; p \mid \ldots \mid \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \phi \rightarrow \phi \mid \phi \leftrightarrow \phi$
$\exists p \, \phi \mid \forall p \, \phi \mid \ldots$

Semantics
Requires a model M : FreeVar(ϕ) $\rightarrow$ {true, false}
Describes when ϕ holds on M ($M \models \phi$)

$M \models p$ iff $M(p) = \text{true}$
$M \models \phi_1 \vee \phi_2$ iff $M \models \phi_1$ or $M \models \phi_2$
…

$M \models \exists p \, \phi$ iff $M' \models \phi$ *for some* $M' \in \{M[p{:=}\text{true}], M[p{:=}\text{false}]\}$
$M \models \forall p \, \phi$ iff $M' \models \phi$ *for every* $M' \in \{M[p{:=}\text{true}], M[p{:=}\text{false}]\}$

Free variables and renaming

Syntax $\phi : \quad p \mid \ldots \mid \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \phi \to \phi \mid \phi \leftrightarrow \phi$

$\exists p\, \phi \mid \forall p\, \phi \mid \ldots$

Free variables and renaming

Syntax $\phi : \; p \mid \ldots \mid \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \phi \to \phi \mid \phi \leftrightarrow \phi$
$\mid \exists p\, \phi \mid \forall p\, \phi \mid \ldots$

Occurrence p is <u>bound</u> in ϕ if it appears under $\exists p$ or $\forall p$
is <u>free</u> in ϕ if it is not bound

Free variables and renaming

Syntax $\phi :\; p \mid \ldots \mid \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \phi \rightarrow \phi \mid \phi \leftrightarrow \phi$

$\exists p\, \phi \mid \forall p\, \phi \mid \ldots$

Occurrence p	is <u>bound</u> in ϕ	if	it appears under $\exists p$ or $\forall p$
	is <u>free</u> in ϕ	if	it is not bound
$\phi(p_1,\ldots,p_k)$	means		all free variables of ϕ are among $p_1,\ldots,p_k$

Free variables and renaming

Syntax $\phi : \; p \mid \ldots \mid \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \phi \rightarrow \phi \mid \phi \leftrightarrow \phi$

$\exists p\, \phi \mid \forall p\, \phi \mid \ldots$

Occurrence p	is <u>bound</u> in ϕ	if	it appears under $\exists p$ or $\forall p$
	is <u>free</u> in ϕ	if	it is not bound
$\phi(p_1,\ldots,p_k)$	means		all free variables of ϕ are among $p_1,\ldots,p_k$
$\phi[p/\alpha]$	denotes		the formula obtained by replacing all *free occurrences* of p by α

Free variables and renaming

Syntax $\phi :\ p \mid \ldots \mid \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \phi \rightarrow \phi \mid \phi \leftrightarrow \phi$

$\exists p\, \phi \mid \forall p\, \phi \mid \ldots$

Occurrence p	is <u>bound</u> in ϕ	if	it appears under $\exists p$ or $\forall p$
	is <u>free</u> in ϕ	if	it is not bound
$\phi(p_1,\ldots,p_k)$	means		all free variables of ϕ are among $p_1,\ldots,p_k$
$\phi[p/\alpha]$	denotes		the formula obtained by replacing all *free occurrences* of p by α

Notation abuse: given $\phi(p)$ $\phi[q]$ is shorthand for $\phi[p/q]$

Free variables and renaming

Occurrence p	is <u>bound</u> in ϕ	if	it appears under $\exists p$ or $\forall p$
	is <u>free</u> in ϕ	if	it is not bound
$\phi(p_1,\ldots,p_k)$	means		all free variables of ϕ are among $p_1,\ldots,p_k$
$\phi[p/\alpha]$	denotes		the formula obtained by replacing all *free occurrences* of p by α

Free variables and renaming

Occurrence p	is bound in ϕ	if	it appears under $\exists p$ or $\forall p$
	is free in ϕ	if	it is not bound
$\phi(p_1,\ldots,p_k)$	means		all free variables of ϕ are among $p_1,\ldots,p_k$
$\phi[p/\alpha]$	denotes		the formula obtained by replacing all *free occurrences* of p by α

Lemma (compositionality)

Let ϕ, α be some QBF and p a variable

Suppose that $M \models p$ iff $M \models \alpha$
Then $M \models \phi$ iff $M \models \phi[p/\alpha]$
provided no free variable of α occurs in ϕ

Free variables and renaming

Occurrence p	is <u>bound</u> in ϕ	if it appears under $\exists p$ or $\forall p$
	is <u>free</u> in ϕ	if it is not bound
$\phi(p_1,\ldots,p_k)$	means	all free variables of ϕ are among $p_1,\ldots,p_k$
$\phi[p/\alpha]$	denotes	the formula obtained by replacing all *free occurrences* of p by α

Lemma (compositionality)

Let ϕ, α be some QBF and p a variable

Suppose that $M \models p$ iff $M \models \alpha$
Then $M \models \phi$ iff $M \models \phi[p/\alpha]$
provided no free variable of α occurs in ϕ

Corollary (renaming)

$\exists p\ \phi$ is equivalent to $\exists q\ \phi[p/q]$
provided q *does not occur in* φ

Economy of syntax

Syntax $\phi : \; p \mid \ldots \mid \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \phi \rightarrow \phi \mid \phi \leftrightarrow \phi$

$\exists p \mid \forall p \mid \ldots$

Lemma $\forall$ can be defined using $\exists, \neg$

Economy of syntax

Syntax ϕ : $p \mid \ldots \mid \phi \vee \phi \mid \phi \wedge \phi \mid \neg\phi \mid \phi \rightarrow \phi \mid \phi \leftrightarrow \phi$

$\exists p \mid \forall p \mid \ldots$

Lemma $\forall$ can be defined using $\exists, \neg$

namely:

$\forall p\, \phi$ is equivalent to $\neg(\exists p\, \neg\phi)$

Algorithms

Model-check(φ, M)

```
if φ = p then
    return M(p)
else if φ = φ1 ∨ φ2 then
    return Model-check(φ1, M) OR
           Model-check(φ2, M)
else if …
…

else if φ = ∃p φ' then
    return Model-check(φ', M[p:=true]) OR
           Model-check(φ', M[p:=false])

else if φ = ∀p φ' then
    return Model-check(φ', M[p:=true]) AND
           Model-check(φ', M[p:=false])
```

Algorithms

Model-check(φ, M)

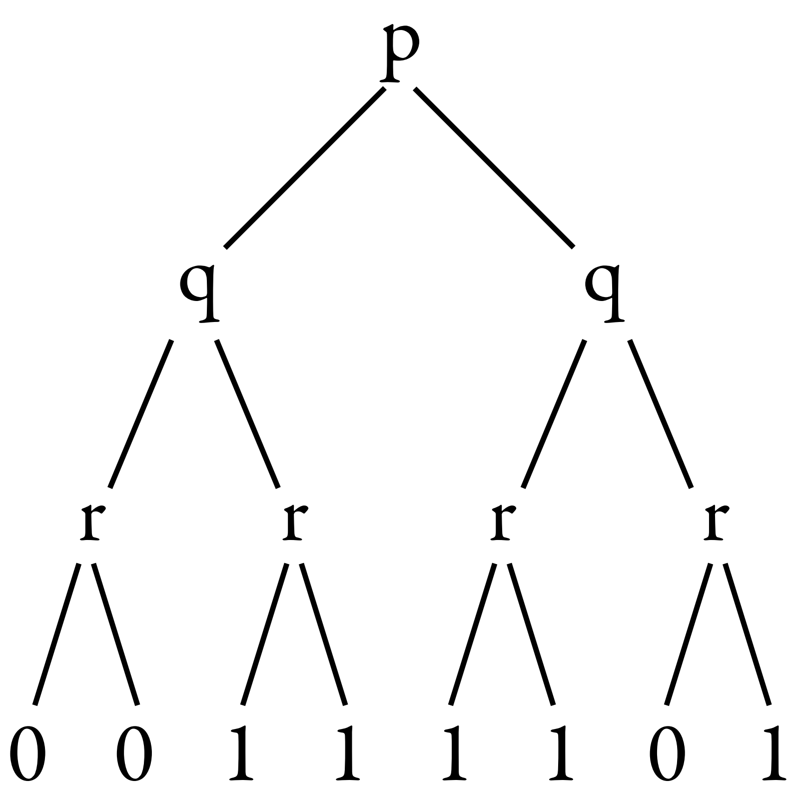
```
if φ = p then
    return M(p)
else if φ = φ1 ∨ φ2 then
    return Model-check(φ1, M) OR
           Model-check(φ2, M)
else if ...
...

else if φ = ∃p φ' then
    return Model-check(φ', M[p:=true]) OR
           Model-check(φ', M[p:=false])

else if φ = ∀p φ' then
    return Model-check(φ', M[p:=true]) AND
           Model-check(φ', M[p:=false])
```

Complexity: **PSPACE**-complete

Propositional satisfiability vs QBF model-checking

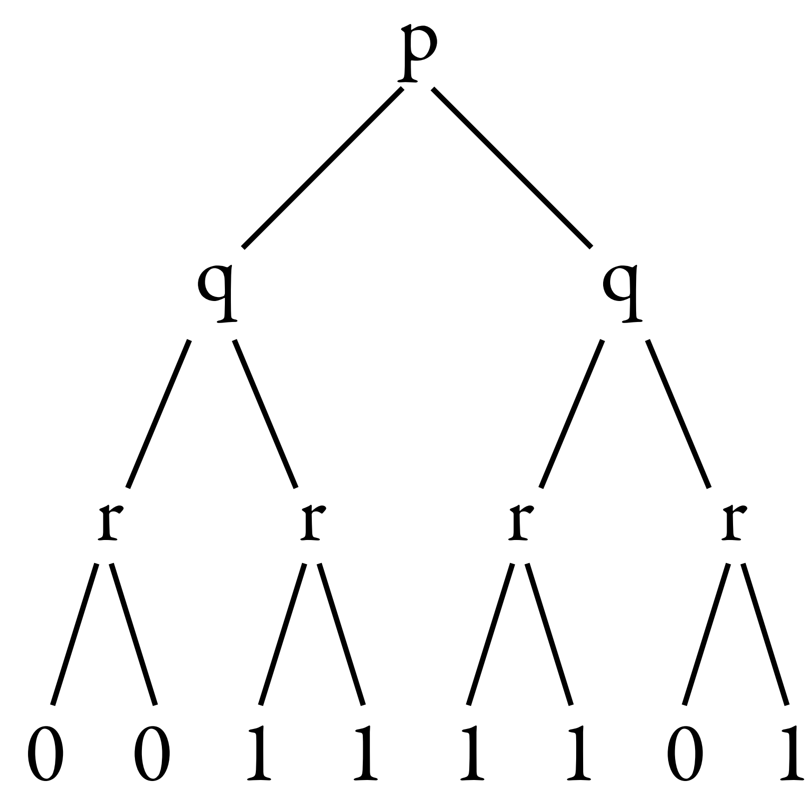
SAT

$[\exists p\ \exists q\ \exists r]\ \phi(p,q,r)$



QBF

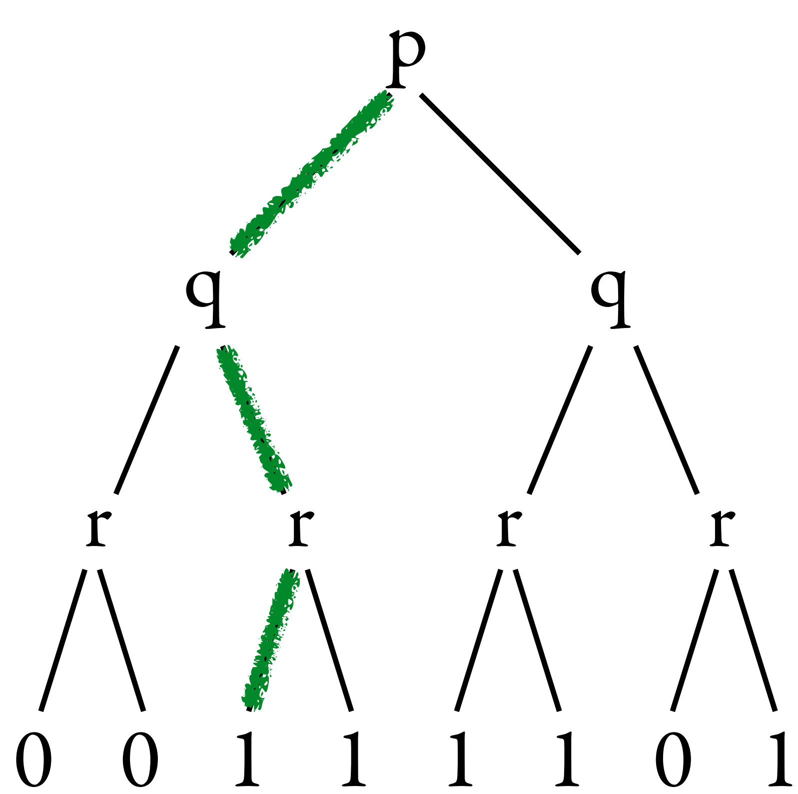
$\forall p\ \exists q\ \forall r\ \phi(p,q,r)$



Propositional satisfiability vs QBF model-checking

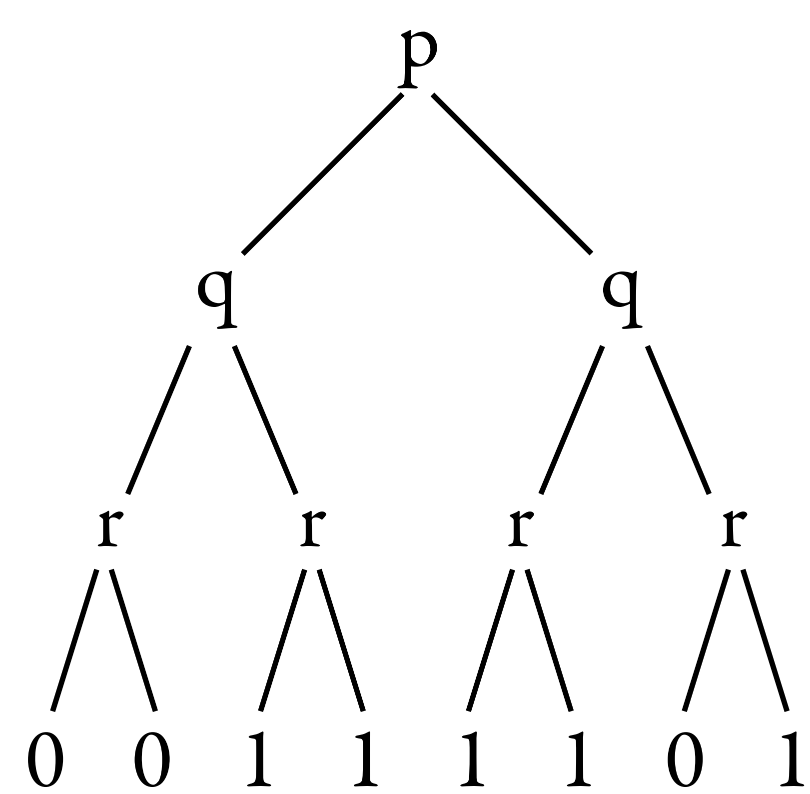
SAT

$[\exists p\ \exists q\ \exists r]\ \phi(p,q,r)$

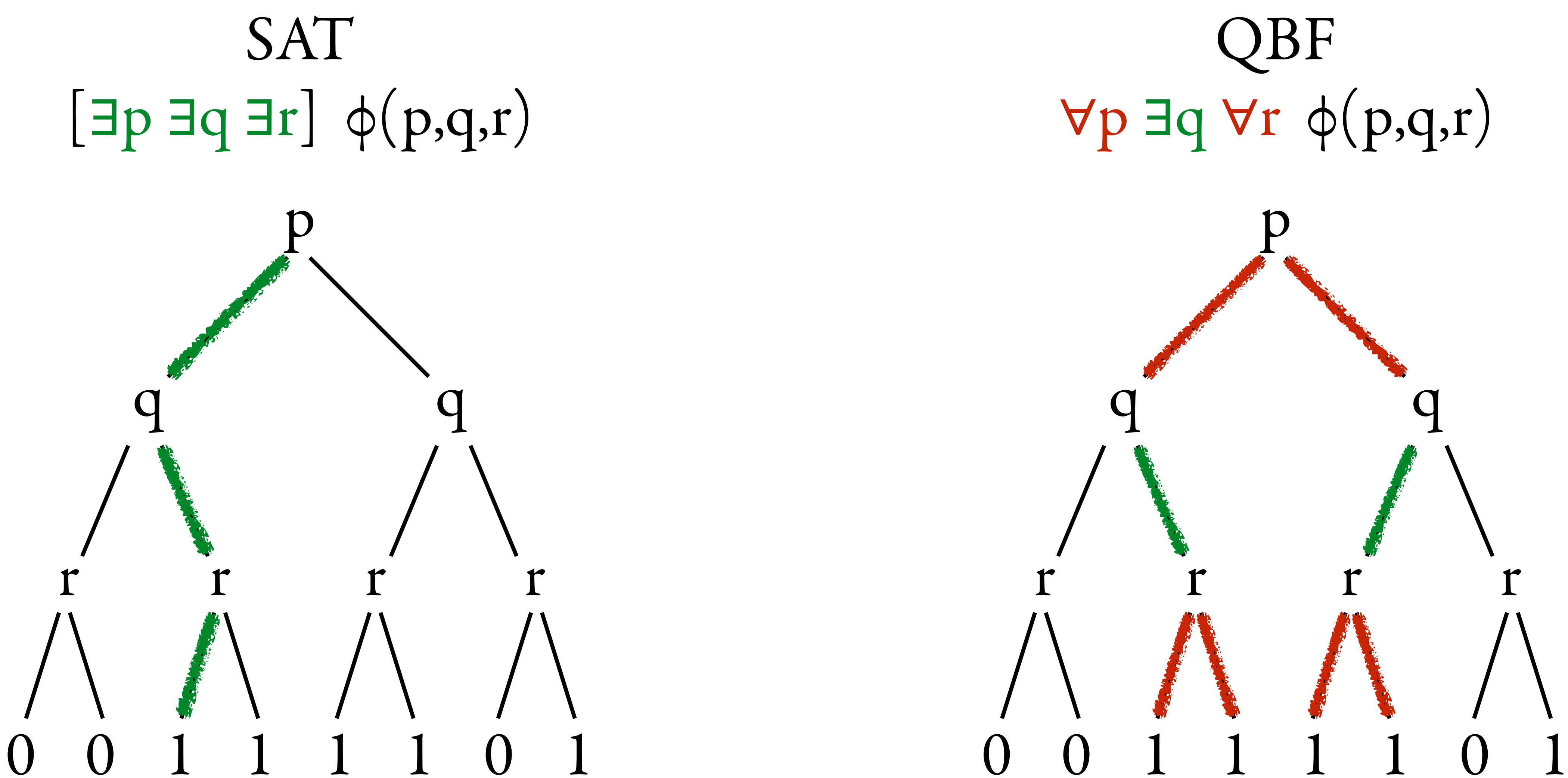


QBF

$\forall p\ \exists q\ \forall r\ \phi(p,q,r)$



Propositional satisfiability vs QBF model-checking



Normal forms

Prenex

$$\phi : \quad \exists p\, \phi \;\mid\; \forall p\, \phi \;\mid\; \alpha$$
$$\alpha : \quad p \;\mid\; \alpha \vee \alpha \;\mid\; \alpha \wedge \alpha \;\mid\; \neg\alpha \;\mid\; \ldots$$

Normal forms

<u>Prenex</u>

$$\phi : \quad \exists p\, \phi \;\mid\; \forall p\, \phi \;\mid\; \alpha$$
$$\alpha : \quad p \;\mid\; \alpha \vee \alpha \;\mid\; \alpha \wedge \alpha \;\mid\; \neg\alpha \;\mid\; \ldots$$

<u>Prenex-CNF/DNF</u>

when in addition α is in CNF/DNF

Normal forms

Prenex

$$\phi : \quad \exists p\, \phi \;\mid\; \forall p\, \phi \;\mid\; \alpha$$
$$\alpha : \quad p \;\mid\; \alpha \vee \alpha \;\mid\; \alpha \wedge \alpha \;\mid\; \neg\alpha \;\mid\; \ldots$$

Prenex-CNF/DNF when in addition α is in CNF/DNF

Lemma 1 Prenex normal form can be computed in polynomial time

Normal forms

Prenex

$\phi : \quad \exists p\, \phi \;\mid\; \forall p\, \phi \;\mid\; \alpha$

$\alpha : \quad p \;\mid\; \alpha \vee \alpha \;\mid\; \alpha \wedge \alpha \;\mid\; \neg\alpha \;\mid\; \ldots$

Prenex-CNF/DNF when in addition α is in CNF/DNF

Lemma 1 Prenex normal form can be computed in polynomial time

Lemma 2 Model-checking prenex QBFs with n quantifier alternations is in

$\Sigma_n = \mathbf{NP}^{\mathbf{coNP}^{\cdots}}$ or $\Pi_n = \mathbf{coNP}^{\mathbf{NP}^{\cdots}}$

Application example — compression

k bits

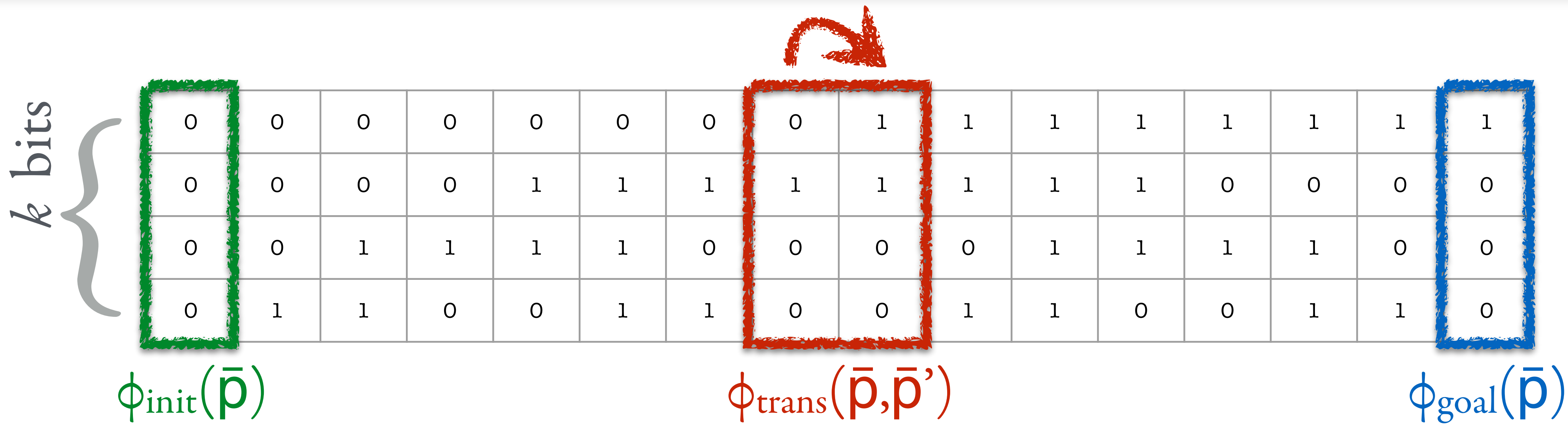
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0

$\phi_{init}(\bar{p})$ $\phi_{trans}(\bar{p},\bar{p}')$ $\phi_{goal}(\bar{p})$

Reachability in $n = 2^k$ steps:

$$\phi_{reach} = \exists \bar{p}_1,\ldots,\bar{p}_n \; \phi_{init}[\bar{p}_1] \wedge \phi_{goal}[\bar{p}_n] \wedge \bigwedge_{i=2,\ldots,n} \phi_{trans}[\bar{p}_{i-1},\bar{p}_i]$$

Application example — compression



Reachability in $n = 2^k$ steps:

$$\phi_{reach} = \exists \bar{p}_1,\ldots,\bar{p}_n \;\; \phi_{init}[\bar{p}_1] \;\wedge\; \phi_{goal}[\bar{p}_n] \;\wedge\; \bigwedge_{i=2,\ldots,n} \phi_{trans}[\bar{p}_{i-1},\bar{p}_i]$$

Size: $|\phi_{reach}| \approx |\phi_{init}| + |\phi_{goal}| + k \cdot 2^k \cdot |\phi_{trans}|$ can we do better?

Application example — compression

k bits

0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0

$\phi_{init}(\bar{p})$ $\phi_{trans}(\bar{p},\bar{p}')$ $\phi_{goal}(\bar{p})$

Reachability in $n = 2^k$ steps:

$$\phi_{reach} = \exists \bar{p}_1,\ldots,\bar{p}_n \; \phi_{init}[\bar{p}_1] \wedge \phi_{goal}[\bar{p}_n] \wedge \bigwedge_{i=2,\ldots,n} \phi_{trans}[\bar{p}_{i-1},\bar{p}_i]$$

$$\forall \bar{s},\bar{s}' \left(\bigvee_{i=2,\ldots,n} \bar{s}\bar{s}' = \bar{p}_{i-1}\bar{p}_i \right) \rightarrow \phi_{trans}[\bar{s},\bar{s}']$$

Size: $|\phi_{reach}| \approx |\phi_{init}| + |\phi_{goal}| + k \cdot 2^k \cdot |\phi_{trans}|$ can we do better?

Application example — compression

k bits

0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0

$\phi_{init}(\bar{p})$ $\phi_{trans}(\bar{p},\bar{p}')$ $\phi_{goal}(\bar{p})$

Reachability in $n = 2^k$ steps:

$$\phi_{reach} = \exists \bar{p}_1,\ldots,\bar{p}_n \;\; \phi_{init}[\bar{p}_1] \wedge \phi_{goal}[\bar{p}_n] \wedge \bigwedge_{i=2,\ldots,n} \phi_{trans}[\bar{p}_{i-1},\bar{p}_i]$$

$$\forall \bar{s},\bar{s}' \left(\bigvee_{i=2,\ldots,n} \bar{s}\bar{s}' = \bar{p}_{i-1}\bar{p}_i \right) \rightarrow \phi_{trans}[\bar{s},\bar{s}']$$

$$k \cdot 2^k + |\phi_{trans}|$$

Size: $|\phi_{reach}| \approx |\phi_{init}| + |\phi_{goal}| + k \cdot 2^k \cdot |\phi_{trans}|$ can we do better?

Application example — compression

k bits

0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0

$\phi_{init}(\bar{p})$ $\phi_{trans}(\bar{p},\bar{p}')$ $\phi_{goal}(\bar{p})$

Reachability in $n = 2^k$ steps:

$$\phi_{reach} = \exists \bar{p}_1,\dots,\bar{p}_n \;\; \phi_{init}[\bar{p}_1] \wedge \phi_{goal}[\bar{p}_n] \wedge \bigwedge_{i=2,\dots,n} \phi_{trans}[\bar{p}_{i-1},\bar{p}_i]$$

$$\forall \bar{s},\bar{s}' \left(\bigvee_{i=2,\dots,n} \bar{s}\bar{s}'=\bar{p}_{i-1}\bar{p}_i \right) \rightarrow \phi_{trans}[\bar{s},\bar{s}']$$

$$\phi_k[\bar{p}_1,\bar{p}_n]$$

$$\phi_1(\bar{p},\bar{q}) = \phi_{trans}[\bar{p},\bar{q}]$$

$$\phi_{i+1}(\bar{p},\bar{q}) = \exists \bar{r}\, \forall \bar{s},\bar{s}'$$
$$\bar{s}\bar{s}'=\bar{p}\bar{r} \vee \bar{s}\bar{s}'=\bar{r}\bar{q} \rightarrow \phi_i[\bar{s},\bar{s}']$$

$$k \cdot 2^k + |\phi_{trans}|$$

Size: $|\phi_{reach}| \approx |\phi_{init}| + |\phi_{goal}| + k \cdot 2^k \cdot |\phi_{trans}|$ can we do better?

Application example — compression

k bits

0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0

$\phi_{init}(\bar{p})$ $\phi_{trans}(\bar{p},\bar{p}')$ $\phi_{goal}(\bar{p})$

Reachability in $n = 2^k$ steps:

$\phi_{reach} = \exists \bar{p}_1,\ldots,\bar{p}_n\ \phi_{init}[\bar{p}_1] \wedge \phi_{goal}[\bar{p}_n] \wedge \bigwedge_{i=2,\ldots,n} \phi_{trans}[\bar{p}_{i-1},\bar{p}_i]$

$\forall \bar{s},\bar{s}' \left(\bigvee_{i=2,\ldots,n} \bar{s}\bar{s}'=\bar{p}_{i-1}\bar{p}_i \right) \rightarrow \phi_{trans}[\bar{s},\bar{s}']$

$\phi_k[\bar{p}_1,\bar{p}_n]$

$\phi_1(\bar{p},\bar{q}) = \phi_{trans}[\bar{p},\bar{q}]$

$\phi_{i+1}(\bar{p},\bar{q}) = \exists \bar{r}\ \forall \bar{s},\bar{s}'$

$\bar{s}\bar{s}'=\bar{p}\bar{r} \vee \bar{s}\bar{s}'=\bar{r}\bar{q} \rightarrow \phi_i[\bar{s},\bar{s}']$

$k^2 + |\phi_{trans}|$

$k \cdot 2^k + |\phi_{trans}|$

Size: $|\phi_{reach}| \approx |\phi_{init}| + |\phi_{goal}| + k \cdot 2^k \cdot |\phi_{trans}|$

can we do better?

Things to remember

Things to remember

- QBF logic is still simple

- QBF model-checking *generalises satisfiability* & *validity* of prop. formulas

  (complexity is slightly higher and depends on quantifier alternation)

- There is a trick to *succinctly describe* a reachability property